



Atty Docket No. 74451.P115

## Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Mail Stop Appeal Brief- Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37(a)**

This is an appeal to the Board of Patent Appeals and Interferences from the decision of the Examiner of Group 2174, dated January 26, 2005, which finally rejected claims 1-40 in the above-identified application. This Appeal Brief is hereby submitted pursuant to 37 C.F.R. § 41.37(a).

**FIRST CLASS CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to Mail Stop Appeal Brief –Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450 on

December 23, 2005

(Date of Deposit)

By Dianne Neathery

(Name of Person Mailing Correspondence)

Dianne Noethen  
Signature

December 23, 2005

Date

## **I. REAL PARTY IN INTEREST**

The real party in interest is the assignee of the full interest in the invention, Ricoh Co., Ltd., 15-5 Minami-Aoyama 1-Chome, Minato-Ku, Tokyo, Japan 107-8544.

## **II. RELATED APPEALS AND INTERFERENCES**

To the best of Appellant's knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision in the instant appeal.

## **III. STATUS OF THE CLAIMS**

Claims 1-40 are pending in the application and were finally rejected in an Office Action mailed January 26, 2005. Claims 1-40 are the subject of this appeal. A copy of Claims 1-40 as they stand on appeal are set forth in Appendix A.

#### **IV. STATUS OF AMENDMENTS**

No amendments have been submitted subsequent to the Final Office Action mailed January 26, 2005.

#### **V. SUMMARY OF CLAIMED SUBJECT MATTER**

Appellant's invention as claimed in claims 1-40 is directed to a method and apparatus for melding user interfaces. Melded user interfaces combines the user interfaces of two or more applications and does not require the cooperation or acquiescence from the applications. Using melded user interfaces, the screen layout (e.g., base layout) corresponding to the user interface of one application may be used by one or more other applications to display data associated with that application.

Independent claim 1 claims a method including: extracting a first data from a display buffer, the first data being generated by a first application and being associated with a user interface from the first application (Specification, pp. 11-13; Fig. 1B); recognizing a layout from the first data (Specification, pp. 11-13 and 14-15; Figs. 1B, 3A-3B, 4A-4C, and 5A-5B); and using the layout to create an overlay to display a second data generated by a second application (Specification, pp. 11-13 and 14-15; Figs. 1B, 3A-3B, 4A-4C, and 5A-5B), where there is no direct link between the first application and the second application and the first data is extracted from the display buffer without cooperation of the first application at runtime (Specification, pp. 9-10). Independent claims 9 and 17 claim the invention as a computer readable medium and a system respectively.

Independent claim 25 claims a method including: modifying data in a display buffer that is generated by a first application with data generated by a second application without cooperation of the first application at runtime, the first application running independently from the second application; and receiving input in response to user interactions with the second application through a user interface associated with the data generated by the first application, where the data generated by the second application is placed in a location in the user interface and the location is contextually consistent with the data generated by the second application (Specification, pp. 9-10, 11-13, and 14-15; Figs. 1B, 3A-3B, 4A-4C, and 5A-5B). Independent claims 29 and 33 claim the invention as a computer readable medium and a system respectively.

Independent claim 37 claims a method including: reading raster data from a raster display buffer containing an image generated by a first application without cooperation of the first application at runtime; performing a pattern recognition on the image to generate a pattern; applying predetermined information about the image with the pattern to determine a layout of the image; generating an overlay using the layout of the image; and placing data generated by a second application on the overlay (Specification, pp. 9-10, 11-13, and 14-15; Figs. 1B, 3A-3B, 4A-4C, and 5A-5B).

## **VI. GROUNDS OF REJECTIONS TO BE REVIEWED ON APPEAL**

- A. Whether claims 1, 9, and 17 are anticipated under 35 U.S.C. §102(b) by U.S. Patent No. 5,596,702 of Stucka et al. (“Stucka”).
- B. Whether claims 2, 10, and 18 are anticipated under 35 U.S.C. §102(b) by Stucka.

C. Whether claims 3-5, 11-13, and 19-21 are anticipated under 35 U.S.C. §102(b) by Stucka.

D. Whether claims 6-7, 14-15, and 22-23 are anticipated under 35 U.S.C. §102(b) by Stucka.

E. Whether claims 8, 16, and 24 are anticipated under 35 U.S.C. §102(b) by Stucka.

F. Whether claims 25, 28-29, 32-33, and 36 are anticipated under 35 U.S.C. §102(b) by Stucka.

G. Whether claims 26, 30, and 34 are anticipated under 35 U.S.C. §102(b) by Stucka.

H. Whether claims 37-40 are anticipated under 35 U.S.C. §102(b) by Stucka.

I. Whether claims 27, 31, and 35 are patentable under 35 U.S.C. §103(a) over Stucka in view of U.S. Patent No. 5,936,625 of Kahl et al. (“Kahl”).

## VII. ARGUMENT

The claims do not stand or fall together.

A. Claims 1, 9, and 17 are not anticipated by Stucka.

Claims 1, 9, and 17 stand or fall together. Claim 1 is the representative claim. As discussed above, Appellant's invention as claimed is directed to melded user interfaces that combine the user interfaces of two or more applications and does not require the cooperation or acquiescence from the applications.

Specifically, independent claim 1 includes limitations of extracting first data from a display buffer, where the first data is generated by a first application, is associated with a user interface from the first application and is extracted without from the display buffer cooperation of the first application at runtime. These limitations are not disclosed or suggested by Stucka.

Stucka discloses a user interface server (UIS) that provides user interface services to multiple applications in a form of a library via a specific API (application programming interface), similar to a software development kit (SDK), which is used by a developer of the respective application. Specifically, Stucka states:

“A user interface server coupled to applications, a display object store, and a window management system. The user interface server allows an application developer to provide an application with a user interface that is independent of any particular window management system.”

(Abstract of Stucka, emphasis added).

Thus, Stucka discloses how an application communicates with the UIS via a set of APIs, which must be compiled and linked (via a compiler and linker of a software development tool, such as a C/C++ compiler and linker) with the application when the application is developed by the developer, to access one or more functions provided by the UIS. See, for example, col. 20, line 33 to col. 21, line 40, and col. 22, line 58 to col. 23, line 55.

However, it is respectfully submitted that Stucka does not disclose the limitation of extracting data of an application from a display buffer, particularly, without cooperation of the application. In fact, there is no mention of extracting data from a display buffer in Stucka.

Even if, for the sake of argument, the access from an application to a UIS server may be considered “extracting data” from a first application (assuming the UIS server is considered as a first application), the access to the UIS cannot be performed via a display buffer. As shown in Figure 3 of Stucka, the UIS is a server that is a dedicated server providing specific services over a network connection (e.g., connection 56 of Figure 3).

UIS 48 of Stucka is not stored in a display buffer 72. Rather, UIS 48, as well as other components, such as working memory area 78, display object store 46, and operating system 74, etc., is stored in a general purpose RAM 38. If Stucka intended to have UIS 48 operating within a display buffer, UIS 48 would have been shown within display buffer 72. There is no suggestion of storing UIS 48 in display buffer 72 throughout Stucka.

One with ordinary skill in the art would not attempt to store, even if it were possible, the UIS (e.g., a server), specifically the library, such as display object store 46, in a display

buffer accessed by another application, such as application 50. Is it respectfully submitted that one with ordinary skill in the art would not believe a server or a library, such as display object store 46 of Stucka, would reasonably be stored in a display buffer.

Even if the communications between applications (50, 53, and 54) and UIS 48 can be considered as extracting data from another application. Such an extraction operation is not performed from a display buffer, particularly display buffer 72. At most, it can only be considered an extraction from RAM 38. See, for example, Fig. 2, col. 7, line 47 to col. 8, line 25 of Stucka. Given the fact that Stucka fails to disclose or suggest that such operations be performed within display buffer 72, it is respectfully submitted that those with ordinary skill in the art would not consider, based on the teachings of Stucka, that the above operations can be performed within a display buffer.

In Stucka, the access to the UIS cannot be performed without cooperation of the UIS. As described above, in order to access UIS, an application has to be compiled and linked with the APIs of the UIS. Without using the APIs of the UIS, an application cannot access the UIS.

In addition, independent claim 1 includes a limitation of recognizing a layout from the extracted first data. It is respectfully submitted that this limitation is also absent from Stucka. Stucka does not disclose or suggest recognizing a layout from the data extracted from a display buffer, which is generated by another application. In fact, there is no mention of recognizing a layout in a display buffer in Stucka.

As discussed above, Stucka relies on a set of APIs used by an application to access a server to retrieve user interface data. There is no need to recognize a layout from the data provided by the UIS, particularly using a pattern recognition operation. The respective

application has to be compiled and linked with the corresponding API, such as the header files and libraries of the UIS, during the development of the application. At run time, when the application calls a specific function (e.g., an API function) to access the server, the server provides services to the application. It is irrelevant whether the application would recognize a layout of the data returned from the server because the application has to call the corresponding function exported and published by the server in a format required by the API, regardless of whether the application recognizes the layout.

In contrast, the present invention as claimed recognizes a layout of a user interface generated by another application in a display buffer (e.g., raster data that is about to be displayed or currently displayed) and overlays its data with the recognized interface, without using cooperation of the application generating the user interface. Therefore, one would not be motivated, based on the teachings of Stucka, to arrive at the present invention as claimed.

Furthermore, independent claim 1 further includes a limitation of using the recognized layout to create an overlay to display a second data generated by a second application, wherein there is no direct link between the first and second applications (e.g., without requiring communications between the first and second applications via APIs). It is respectfully submitted that this limitation is also absent from the cited references, individually or in combination.

As discussed above, an application of Stucka has to rely on a set of APIs (e.g., cooperation) to access the UIS in order to create a user interface. Nowhere in Stucka disclose or suggest an operation of creating an overlay of a recognized layout when an application accesses the UIS via a set of APIs. There is no need or motive to create an overlay to display the data from another application, as discussed above.

Further, contrary to the present invention as claimed, it is respectfully submitted that there is a direct link between the application and the UIS in Stucka. The source code of an application in Stucka must be compiled and linked with the exported or published APIs from the UIS. Any changes in the APIs would cause the communications between the application and the UIS to fail, which requires either the application or the UIS, or the both to be recompiled and linked. The applications and the UIS of Stucka depend on a set of APIs mutually agreed upon, which teaches away from the present invention as claimed. See, for example, col. 23, lines 38 to 54. Therefore, there is a direct link between an application and the UIS in Stucka. In contrast, there is no direct link between the first and second applications in the present invention as claimed, where the data is extracted without using an API of the first application.

In the Office Action, the Examiner stated:

“As per independent claim 1, Stucka et al. teaches a method comprising:

Extracting a first data from a display buffer, the first data generated by a first application and being associated with a user interface from the first application; (col 23 lines 62-67, col 24 lines 37-60)

Recognizing a layout from the first data; and

Using the layout to create an overlay to display a second data generated by a second application (col 26, lines 66-67, col 27, lines 1-5), wherein there is no direct link between the first application and the second application (col 4, lines 64-67, col 5, lines 1-2);

And wherein the first data is extracted from the display buffer without cooperation of the first application at runtime (col. 10, lines 44-68; Since the setting information of the user's interface is stored to a server and load from the server, therefore it is inherent that cooperation of first application at runtime)”

(1/26/2005 Office Action, pages 2-3, emphasis added).

Appellant respectfully disagrees for the reasons set forth above. As discussed above, the system RAM of Stucka is not and cannot be considered as a display buffer. Specifically, for example, a section of Stucka relied upon by the Examiner states:

“In step 701, application A issues an initialize command. The initialize command allows the application to set up parameters for the exchange of information with the user interface server. The initialize command is sent to the UIS with various command parameter data required to initialize the user interface sessions between application A and the UIS. The initialize command itself may be utilized multiple times by the application to establish multiple user interface sessions. In this example, however, a single user interface session will be utilized to demonstrate the invention. The initialize command can also specify certain default parameters concerning user interface components. For example, background and foreground colors could be specified. Those of ordinary skill in the art would be able to modify the initialize command so that a single command, with appropriate parameters, would initialize multiple user interface sessions.

In step 703, the UIS receives the initialize command and initializes appropriate UIS parameters such as working memory areas and other internal variables. Step 701 may include establishing a connection with the WMS to enable communication between the UIS and the WMS for this particular user interface session. Thus the UIS may send an initialize command to the WMS to establish appropriate window management session for each user interface desired by the application. As stated previously, included with the initialize command sent from the application may be default values for resources common to all components or sets of components such as background, foreground, color, text, font, etc.”

(Stucka, col. 23 line 64 to col. 24, line 60).

Appellant respectfully submits that the above section does not use a display buffer, such as display buffer 72 shown in Fig. 2 of Stucka. Clearly, Stucka’s RAM 38 and working memory area 78 are not a display buffer. Given the fact that Stucka fails to disclose or suggest such operations be performed within display buffer 72, it is respectfully submitted that Stucka teaches away from the present invention as claimed and those with ordinary skill in the art would not consider, based on the teachings of Stucka, that the above operations can be performed within a display buffer.

Further, it appears that the Examiner considers RAM 38 or working memory area is a display buffer. Specifically, Examiner stated:

“Contrary to Applicant’s assertion, working RAM, specifically dedicated to display operations, as disclosed by Stucka, is commonly used as a display buffer.

Consider embedded system in which the video card does not have on-board RAM, of necessity, the working RAM must be used for a display buffer. In fact, there was the standard operation before on-board RAM was integrated into video cards.

...

Therefore, it is logical to conclude that when system extracts the component and layout of a interface from a working memory or WMA, (fig. 6c), that systems is in fact extracting data from a display buffer or any elements that are showed on figure 2.”

(1/26/2005 Office Action, pages 9-10, emphasis added).

Appellant respectfully disagrees. Appellant respectfully submits that those with ordinary skill in the art would not consider RAM 38 is a display buffer for storing user interface server 48, window system I/F 70, window management system 58, display object store 46, operating system 74, etc. Rather, Stucka designates a display buffer 72 for display purposes (Fig. 2 of Stucka).

In addition, it appears that the Examiner admitted that “since the setting information of the user’s interface is stored to a server and load from the server, therefore it is inherent that cooperation of first application at runtime” (see the section of the Office Action cited above, emphasis added), while independent claim 1 clearly requires that there is no cooperation between two applications at runtime. Thus, the Examiner admitted that Stucka fails to disclose each and every limitation of independent claim 1.

Further, the Examiner’s embedded system interpretation was not based on Stucka’s disclosure. Rather, it appears that the Examiner’s interpretation was based on the hindsight of Appellant’s disclosure. It would be impermissible hindsight to use Appellant’s own disclosure against the Appellant.

In order to anticipate a claim, each and every limitation of the claim must be taught by Stucka. It is respectfully submitted that Stucka fails to teach each and every limitation of claim 1. Therefore, for the reasons discussed above, independent claim 1 is not anticipated by Stucka. Similarly, independent claims 9 and 17 include limitations similar to those discussed above. Similar arguments with respect to claim 1 are applied herein to claims 9 and 17. Therefore, for reasons similar to those discussed above, independent claims 9 and 17 are not anticipated by Stucka.

B. Claims 2, 10, and 18 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 2, 10, and 18 stand or fall together. Claim 2 is a representative claim. Claims 2, 10, and 18 depend from, directly or indirectly, independent claims 1, 9, and 17, respectively. The reasons cited above with respect to claims 1, 9, and 17 are applicable to claims 2, 10, and 18 and are herein incorporated by reference. Based on at least these reasons, claims 2, 10, and 18 are not anticipated by Stucka.

In addition, for example, claim 2 requires performing a pattern recognition operation on the first data to create the layout. It is respectfully submitted that these limitations are absent from Stucka. The Examiner contended that col. 23 line 64 to col. 24, line 60 of Stucka discloses such limitations (1/26/2005 Office Action, page 3). Appellant respectfully disagrees. The cited section, as discussed above, does not disclose/suggest performing recognition operation on the first data to create the layout. Rather, the cited section merely describes how one application accesses the server and/or library via a set of APIs, instead of

manipulating data in the display buffer as claimed in the present application. Similar limitations of claim 2 appear in claims 10 and 18. It is respectfully submitted that Stucka fails to disclose each and every limitations of claims 2, 10, and 18.

Therefore, in addition to the reasons applied to their respective independent claims, claims 2, 10, and 18 are independently not anticipated by Stucka.

C. Claims 3-5, 11-13, and 19-21 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 3-5, 11-13, and 19-21 stand or fall together. Claims 3-5, 11-13, and 19-21 depend from, directly or indirectly, independent claims 1, 9, and 17, respectively. The reasons cited above with respect to claims 1, 9, and 17 are applicable to claims 3-5, 11-13, and 19-21 and are herein incorporated by reference. Based on at least these reasons, claims 3-5, 11-13, and 19-21 are not anticipated by Stucka.

In addition, claims 3-5, 11-13, and 19-21 require determining an overlay location of the layout to place the second data based on some information about the layout, generating the overlay of the layout and place the second data in the overlay, and merging the overlay with the layout. It is respectfully submitted that these limitations are also absent from Stucka.

The Examiner contended that col. 26, line 66 to col. 27, line 5 of Stucka discloses such limitations (1/26/2005 Office Action, page 3).

Appellant respectfully disagrees. Again, the cited section of Stucka is related to accessing the UIS via a set of APIs, instead of manipulating data in the display buffer as

claimed in the present application. It is respectfully submitted that Stucka fails to disclose each and every limitations of claims 3-5, 11-13, and 19-21.

Therefore, in addition to the reasons applied to their respective base claims, claims 3-5, 11-13, and 19-21 are independently not anticipated by Stucka.

D. Claims 6-7, 14-15, and 22-23 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 6-7, 14-15, and 22-23 stand or fall together. Claims 6-7, 14-15, and 22-23 depend from, directly or indirectly, independent claims 1, 9, and 17, respectively. The reasons cited above with respect to claims 1, 9, and 17 are applicable to claims 6-7, 14-15, and 22-23 and are herein incorporated by reference. Based on at least these reasons, claims 6-7, 14-15, and 22-23 are not anticipated by Stucka.

In addition, claims 6-7, 14-15, and 22-23 include writing the overlay in a display buffer such that the second data is displayed at the overly location without changing sections of the first data outside of the overlay location and interacting with the second application via the second data at the overlay location. It is respectfully submitted that these limitations are absent from Stucka.

The Examiner contended that sections of col. 23, line 62 to col. 24, line 60 and col. 26, line 66 to col. 27 line 5 of Stucka disclose such limitations (1/26/2005 Office Action, pages 3-4). Appellant respectfully disagrees. Again, the cited sections of Stucka are related to accessing the UIS via a set of APIs, instead of manipulating data in the display buffer as claimed in the present application. It is respectfully submitted that Stucka fails to disclose

each and every limitations of claims 6-7, 14-15, and 22-23.

Therefore, in addition to the reasons applied to their respective base claims, claims 6-7, 14-15, and 22-23 are independently not anticipated by Stucka.

E. Claims 8, 16, and 24 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 8, 16, and 24 stand or fall together. Claims 8, 16, and 24 depend from, directly or indirectly, independent claims 1, 9, and 17, respectively. The reasons cited above with respect to claims 1, 9, and 17 are applicable to claims 8, 16, and 24 and are herein incorporated by reference. Based on at least these reasons, claims 8, 16, and 24 are not anticipated by Stucka.

In addition, claims 8, 16, and 24 require that both applications run independently. It is respectfully submitted that this limitation is absent from Stucka. As discussed above, Stucka requires an application to be compiled and linked with the set of libraries in order to allow the application to access UIS. Thus, the application and the UIS cannot be run independently. Rather, the application has to call the UIS via a set of APIs. It is respectfully submitted that Stucka fails to disclose each and every limitation of claims 8, 16, and 24.

Therefore, in addition to the reasons applied to their respective base claims, claims 8, 16, and 24 are independently not anticipated by Stucka.

F. Claims 25, 28-29, 32-33, and 36 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 25, 28-29, 32-33, and 36 stand or fall together. Claim 25 is a representative claim. Claim 25 requires modifying data in a display buffer that is generated from a first application with data generated by a second application without cooperation of the first application at runtime, where the first and second applications run independently. In response to user interactions with the second application, an input is received via a user interface associated with the data generated by the first application, where the data generated by the second application is placed in a location of the user interface and the location is contextually consistent with the data generated by the second application. It is respectfully submitted that these limitations are absent from Stucka.

The Examiner rejected claims 25, 28-29, 32-33, and 36 citing similar sections (e.g., col. 23, line 62 to col. 24, line 60; col. 26, line 66 to col. 27, line 5; 1/26/2005 Office Action, pages 5-6) and similar reasons with respect to claims 1, 9, and 17. Appellant respectfully disagrees based on the reasons similar to those set forth above. Therefore, for at least the reasons similar to those with respect to claims 1, 9, and 17, it is respectfully submitted that claims 25, 28-29, 32-33, and 36 are not anticipated by Stucka.

G. Claims 26, 30, and 34 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 26, 30, and 34 stand or fall together. Claims 26, 30, and 34 depend from, directly or indirectly, independent claims 25, 29, and 33 respectively. The reasons cited above with respect to claims 25, 29, and 33 are applicable to claims 26, 30, and 34 and are herein incorporated by reference. Based on at least these reasons, claims 26, 30, and 34 are not anticipated by Stucka.

In addition, claims 26, 30, and 34 include performing a pattern recognition operation on the data generated from the first application to create a layout and forming an overlay with the layout and the predetermined information about a display corresponding to the user interface, where the overlay is utilized to determine placement of the data generated by the second application in the display. It is respectfully submitted that these limitations are also absent from Stucka.

The Examiner rejected claims 26, 30, and 34 citing similar sections (e.g., col. 23, line 62 to col. 24, line 60; col. 26, line 66 to col. 27, line 5; 1/26/2005 Office Action, pages 6-7) and similar reasons with respect to claims 25, 29, and 33. Appellant respectfully disagrees based on the reasons similar to those set forth above. Therefore, for at least the reasons similar to those with respect to claims 25, 29, and 33, it is respectfully submitted that claims 26, 30, and 34 are not anticipated by Stucka.

H. Claims 37-40 are not anticipated under 35 U.S.C. §102(b) by Stucka.

Claims 37-40 stand or fall together. Claim 37 is the representative claim. Claim 37 includes reading raster data from a raster display buffer having an image generated by a first

application without cooperation of the first application at runtime, performing a pattern recognition on the image to generate a pattern, applying predetermined information about the image with the pattern to determine the layout of the image, generating an overlay using the layout of the image, and placing data generated by a second application on the overlay. It is respectfully submitted that these limitations are absent from Stucka.

The Examiner rejected claims 37-40 citing similar sections (e.g., col. 23, line 62 to col. 24, line 60; col. 26, line 66 to col. 27, line 5; 1/26/2005 Office Action, pages 7-8) and similar reasons with respect to claims 1, 9, and 17. Appellant respectfully disagrees based on the reasons set forth above.

In addition, the Examiner also admitted that the applications of Stucka require cooperation from each other, while claim 37 sets forth that both applications do not need cooperation at runtime (1/26/2005 Office Action, pages 7-8).

Therefore, for at least the reasons similar to those with respect to claims 1, 9, and 17, it is respectfully submitted that claims 37-40 are not anticipated by Stucka.

I. Claims 27, 31, and 35 are patentable under 35 U.S.C. §103(a) over Stucka in view of Kahl.

Claims 27, 31, and 35 stand or fall together. Claims 27, 31, and 35 depend from, directly or indirectly, independent claims 25, 29, and 33 respectively. The reasons cited above with respect to claims 25, 29, and 33 are applicable to claims 27, 31, and 35 and are

herein incorporated by reference. Based on at least these reasons, claims 27, 31, and 35 are not anticipated by and are patentable over Stucka.

In addition, claims 27, 31, and 35 include limitations that the layout includes grid cells corresponding to the display areas in the user interface and the data generated from the second application is placed in the grid cells. It is respectfully submitted that these limitations are absent from Stucka and Kahl individually or in combination.

Examiner contended that Fig. 3 of Kahl teaches a graphical user interface layout having grid cells corresponding to display areas in the user interface (1/26/2005 Office Action, page 9). Appellant respectfully disagrees. Fig. 3 of Kahl is directed to a computer system, contrary to the Examiner's statements in the Office Action. Although Kahl discloses a calendar, Kahl still fails to disclose the limitations set forth in claim 27, particularly, for overlay user interface purposes.

Therefore, in addition to the reasons applied to their respective independent claims, claims 27, 31, and 35 are independently patentable over Stucka in view of Kahl. Withdrawal of the rejections is respectfully requested.

## VIII. CONCLUSION

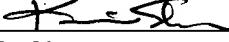
For the reasons stated above, claims 1-40 are not anticipated under 35 U.S.C. §102(b) by Stucka and are patentable under 35 U.S.C. § 103(a) over Stucka in view of Kahl. Appellant respectfully requests that the Board reverse the rejections of the claims 1-40 and direct the Examiner to enter a Notice of Allowance for claims 1-40.

Enclosed is a check in the amount of \$500.00 to cover the fee for filing a brief in support of an appeal as required under 37 C.F.R. § 1.17(c) and 41.20(b)(2).

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due. Furthermore, if an extension is required, then Appellant hereby requests such extension.

Respectfully submitted,  
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: December 23, 2005

  
Kevin G. Shao  
Attorney for Appellant  
Registration No. 45,095

Customer No. 008791  
12400 Wilshire Boulevard  
Seventh Floor  
Los Angeles, CA 90025-1030  
(408) 720-8300

**APPENDIX A: Claims on Appeal**

(37 C.F.R. § 41.37(c)(1)(viii))

The claims on appeal read as follows:

1. (Previously presented) A method comprising:

extracting a first data from a display buffer, the first data being generated by a first application and being associated with a user interface from the first application; recognizing a layout from the first data; and using the layout to create an overlay to display a second data generated by a second application, wherein there is no direct link between the first application and the second application, and wherein the first data is extracted from the display buffer without cooperation of the first application at runtime.

2. (Original) The method of claim 1, wherein recognizing the layout comprises performing a pattern recognition operation on the first data to create the layout.

3. (Original) The method of claim 1, wherein using the layout to create the overlay comprises:

determining an overlay location on the layout to place the second data based on known information about the layout; generating the overlay of the layout; placing the second data in the overlay; and merging the overlay with the layout.

4. (Original) The method of claim 3, wherein the overlay location has a context consistent with the second data.
5. (Original) The method of claim 4, wherein the context is provided by the first application, and wherein a user interacts with the second application using the context.
6. (Original) The method of claim 1, further comprising:
  - writing the overlay in the display buffer such that the second data is displayed at the overlay location without changing sections of the first data outside of the overlay location;
  - displaying information in the display buffer; and
  - interacting with the second application through the second data at the overlay location.
7. (Original) The method of claim 6, further comprising running the first application in the background while interacting with the second application.
8. (Original) The method of claim 1, wherein the first application runs independently from the second application.
9. (Previously presented) A machine-readable medium providing instructions, which when executed by a set of one or more processors, cause said set of processors to perform the following:
  - extracting a first data from a display buffer, the first data being generated by a first

application and being associated with a user interface from the first application;  
recognizing a layout from the first data; and  
using the layout to create an overlay to display a second data generated by a second  
application, wherein there is no direct link between the first application and the second  
application, and wherein the first data is extracted from the display buffer without  
cooperation of the first application at runtime.

10. (Original) The machine-readable medium of claim 9, wherein recognizing the layout  
comprises performing a pattern recognition operation on the first data to create the layout.

11. (Original) The machine-readable medium of claim 9, wherein using the layout to  
create the overlay comprises:

    determining an overlay location on the layout to place the second data based on  
    known information about the layout;  
    generating the overlay of the layout;  
    placing the second data in the overlay; and  
    merging the overlay with the layout.

12. (Original) The machine-readable medium of claim 11, wherein the overlay location  
has a context consistent with the second data.

13. (Original) The machine-readable medium of claim 12, wherein the context is provided by the first application, and wherein a user interacts with the second application using the context.

14. (Original) The machine-readable medium of claim 9, further comprising:  
writing the overlay in the display buffer such that the second data is displayed at the overlay location without changing sections of the first data outside of the overlay location;  
displaying information in the display buffer; and  
interacting with the second application through the second data at the overlay location.

15. (Original) The machine-readable medium of claim 14, further comprising running the first application in the background while interacting with the second application.

16. (Original) The machine-readable medium of claim 9, wherein the first application runs independently from the second application.

17. (Previously presented) A computer system, comprising:  
a bus;  
a data storage device coupled to the bus; and  
a processor coupled to the data storage device, the processor operable to receive instructions which, when executed by the processor, cause the processor to perform a method comprising:

extracting a first data from a display buffer, the first data being generated by a first application and being associated with a user interface from the first application; recognizing a layout from the first data; and using the layout to create an overlay to display a second data generated by a second application, wherein there is no direct link between the first application and the second application, and wherein the first data is extracted from the display buffer without cooperation of the first application at runtime.

18. (Original) The system of claim 17, wherein recognizing the layout comprises performing a pattern recognition operation on the first data to create the layout.

19. (Original) The system of claim 17, wherein using the layout to create the overlay comprises:

determining an overlay location on the layout to place the second data based on known information about the layout;

generating the overlay of the layout;

placing the second data in the overlay; and

merging the overlay with the layout.

20. (Original) The system of claim 19, wherein the overlay location has a context consistent with the second data.

21. (Original) The system of claim 20, wherein the context is provided by the first application, and wherein a user interacts with the second application using the context.
22. (Original) The system of claim 17, further comprising:
  - writing the overlay in the display buffer such that the second data is displayed at the overlay location without changing sections of the first data outside of the overlay location;
  - displaying information in the display buffer; and
  - interacting with the second application through the second data at the overlay location.
23. (Original) The system of claim 22, further comprising running the first application in the background while interacting with the second application.
24. (Original) The system of claim 17, wherein the first application runs independently from the second application.
25. (Previously presented) A method, comprising:
  - modifying data in a display buffer that is generated by a first application with data generated by a second application without cooperation of the first application at runtime, the first application running independently from the second application; and
  - receiving input in response to user interactions with the second application through a user interface associated with the data generated by the first application, wherein the data generated by the second application is placed in a location in the user interface, wherein the

location is contextually consistent with the data generated by the second application.

26. (Original) The method of claim 25, wherein modifying data in the display buffer comprises:

performing a pattern recognition operation on the data generated by the first application to create a layout; and  
forming an overlay with the layout and with predetermined information about a display corresponding to the user interface, the overlay used to determine placement of the data generated by the second application in the display.

27. (Original) The method of claim 26, wherein the layout comprises of grid cells corresponding to display areas in the user interface, and wherein the data generated by the second application is placed in the grid cells.

28. (Original) The method of claim 25, wherein the first application runs in the background while the user interacts with the second application.

29. (Previously presented) A machine-readable medium providing instructions, which when executed by a set of one or more processors, cause said set of processors to perform the following:

modifying data in a display buffer that is generated by a first application with data generated by a second application without cooperation of the first application at runtime, the first application running independently from the second application; and

receiving input in response to user interactions with the second application through a user interface associated with the data generated by the first application, wherein the data generated by the second application is placed in a location in the user interface, wherein the location is contextually consistent with the data generated by the second application.

30. (Original) The machine-readable medium of claim 29, wherein modifying data in the display buffer comprises:

performing a pattern recognition operation on the data generated by the first application to create a layout; and

forming an overlay with the layout and with predetermined information about a display corresponding to the user interface, the overlay used to determine placement of the data generated by the second application in the display.

31. (Original) The machine-readable medium of claim 30, wherein the layout comprises of grid cells corresponding to display areas in the user interface, and wherein the data generated by the second application is placed in the grid cells.

32. (Original) The machine-readable medium of claim 29, wherein the first application runs in the background while the user interacts with the second application.

33. (Previously presented) A computer system, comprising:

a bus;  
a data storage device coupled to the bus; and  
a processor coupled to the data storage device, the processor operable to receive instructions which, when executed by the processor, cause the processor to perform a method comprising:

modifying data in a display buffer that is generated by a first application with data generated by a second application without cooperation of the first application at runtime, the first application running independently from the second application; and

receiving input in response to user interactions with the second application through a user interface associated with the data generated by the first application, wherein the data generated by the second application is placed in a location in the user interface, wherein the location is contextually consistent with the data generated by the second application.

34. (Original) The computer system of claim 33, wherein modifying data in the display buffer comprises:

performing a pattern recognition operation on the data generated by the first application to create a layout; and

forming an overlay with the layout and with predetermined information about a display corresponding to the user interface, the overlay used to determine placement of the data generated by the second application in the display.

35. (Original) The computer system of claim 34, wherein the layout comprises of grid cells corresponding to display areas in the user interface, and wherein the data generated by the second application is placed in the grid cells.

36. (Original) The computer system of claim 33, wherein the first application runs in the background while the user interacts with the second application.

37. (Previously presented) A method comprising:

reading raster data from a raster display buffer containing an image generated by a first application without cooperation of the first application at runtime;

performing a pattern recognition on the image to generate a pattern;

applying predetermined information about the image with the pattern to determine a layout of the image;

generating an overlay using the layout of the image; and

placing data generated by a second application on the overlay.

38. (Original) The method of claim 37, further comprising writing the overlay into the raster display buffer.

39. (Original) The method of claim 37, wherein the image comprises a user interface from the first application, and wherein a user interacts with the second application through the user interface while the first application runs in the background.

40. (Original) The method of claim 39, wherein while the user interacts with the second application, the first application has no control of input received from the user.